

Rad sa Tkinter kontrolama

U dosadašnjem radu sa Tkinterom već smo se upoznali sa nekim kontrolama, a u ovoj nastavnoj jedinici ćemo obraditi dodatne kontrole koje se često sreću u svakodnevnom radu sa GUI aplikacijama.

Neki od najčešćih argumenata koji su dostupni skoro svim kontrolama su: background, foreground, font, cursor, padx, pady, width, height, borderwidth i text.

Button

Ovu kontrolu koristimo kada želimo da omogućimo direktnu interakciju korisnika sa programom. Samo dugme može sadržati tekst, ali i slike (ikonice). Tkinter takođe podržava izvršavanje predefinisane funkcije nakon što korisnik klikne na dugme. Ovom konstruktoru možemo proslediti više argumenata, a najbitniji su master – koji se odnosi na kontejnerski objekat; text – koji se odnosi na tekst koji će biti ispisan na dugmetu, kao i command – argument kojem se prosleđuje ime funkcije koja će se izvršiti nakon klika. Ostatak argumenata se može naći u zvaničnoj [dokumentaciji](#). Možemo napraviti primer gde se vrednost labela uvek ažurira nakon što korisnik klikne na dugme:

Primer upotrebe Button kontrole

```
import tkinter as tk
counter = 0

def button_func():
    global counter
    counter += 1
    label_string.set("Button clicked {} times".format(str(counter)))

win = tk.Tk()
win.minsize(300, 350)
label_string = tk.StringVar(value = "Button clicked {}
times".format(str(counter)))

tk.Label(master = win, textvariable = label_string).pack()
tk.Button(master = win, text = 'ClickMe!', command = button_func).pack()
win.mainloop()
```

Da bismo rešili problem gde funkcija koja će se pozvati nakon klika na dugme *pamti* broj klika, koristimo globalnu promenljivu counter, čiju vrednost uvećavamo za 1 kad god se button_func() pozove. Ta funkcija se prosleđuje konstruktoru tk.Button() klase argumentom command.

Izmenite kod tako da ekran bude dimenzije 200x200 i da sadrži dugme koje ispisuje Hello World u komandnom prozoru.

Text vidžet

Text vidžet je sličan Entry kontroli, sa bitnom razlikom da je u Text vidžetu moguće uneti tekst sa više linija.

Možemo iskoristiti prethodni primer i, pored labele, vrednost promenljive counter zajedno sa pratećim tekstom upisati u Text kontrolu. Takođe je moguće postaviti sliku u ovaj vidžet pored teksta, ubacivati direktne linkove (hiperlinkove) kao i menjati font teksta unutar ovog vidžeta na nivou reči. Za razliku od drugih kontrola, Text vidžet se ne može povezati sa StringVar promenljivama, pa se sadržajem mora rukovati metodama get(), insert() i delete(). Princip rada sa tim metodama se zasniva na tome da se prosleđuje ili jedan ili dva indeksa – koji predstavljaju raspon selekcije teksta unutar kontrola nad kojima želimo da primenimo metodu. Ovi indeksi se prosleđuju kao stringovi i imaju više formata:

- prosleđuju se broj linije i broj kolone, odvojeni tačkom: 1.0, 1.1; linije počinju od broja 1, a kolone od 0, pa tako prvi karakter u prvoj liniji ima indeks: 1.0;
- ugrađena string konstanta tk.END, koja označava kraj kontrole (poslednji karakter poslednje linije);
- numerički indeks koji se prosleđuje kao string iz prve tačke plus neka od sledećih komandi, isto tipa string: linestart – početak linije, lineend – kraj linije, wordstart – početak reči, kao i wordend – kraj reči; na primer, prosleđeni string 2.0 lineend znači kraj treće linije, dok se wordstart/wordend stringovi koriste kada želimo da nađemo početak reči čiji smo indeks karaktera prosledili: ako imamo string Hello World u Text kontroli i prosledimo metodu get('1.7 wordstart'), dobićemo slovo W, koje je početno slovo (wordstart) reči *World*;
- plus minus operatori zajedno sa indeksima i string konstantama; u primeru gde je string Hello World, ako bismo želeli da dobavimo preposlednje slovo prve reči, prosledili bismo: '1.1 wordend - 1 chars'.

Za potrebe našeg primera, prethodni kod koji je koristio labelu treba izmeniti da izgleda ovako:

Primer upotrebe Text kontrole

```
import tkinter as tk
counter = 0

def button_func(text_widget):
    global counter
    counter +=1
    text_widget.delete("1.0", tk.END)
    text_widget.insert("1.0", "Button clicked {} times".format(str(counter)))

win = tk.Tk()
win.minsize(300, 350)

text_widget = tk.Text(master = win)
text_widget.pack()
tk.Button(master = win, text = 'ClickMe!', command = lambda:
button_func(text_widget)).pack()
win.mainloop()
```

Prva od razlika u ovom primeru je da sada više ne koristimo labelu već tk.Text() vidžet. Zbog načina na koji pack() metoda funkcioniše, veći deo ekrana će nam biti pokriven ovim vidžetom, dok će se tek pri dnu, preko čitave širine, prikazati dugme. Konstruktor tk.Button() se takođe razlikuje od primera sa labelom, jer umesto da prosledimo samo funkciju button_func(), prosleđujemo zapravo lambda funkciju koja poziva button_func(). Ovo smo učinili zato što je button_func() funkciji potrebno proslediti i argument. Argument u ovom slučaju predstavlja objekat tk.Text() kontrole u koju će button_func() upisati vrednost globalne promenljive counter.

Sama button_func() funkcija je ispisana tako da se prvo briše sav tekst iz vidžeta, a onda u prvu liniju upisuje trenutna vrednost promenljive counter. Brisanje iz kontrole smo izvršili naredbom: text_widget.delete("1.0", tk.END) – dakle, obrisati sve od prve linije nulte kolone do kraja te linije, a upisivanje smo izvršili naredbom text_widget.insert("1.0", "Button clicked {} times\n".format(str(counter))) – dakle, upisujemo zadati string od prvog reda nulte kolone.

Pitanje

Koji argument koristimo prilikom inicijalizacije tk.Button kontrole ako želimo da nam se nakon klika na to dugme izvrši funkcija?

- function
- **command**
- func

Objašnjenje:

Tačan odgovor je da koristimo argument command, kome prosleđujemo funkciju koju želimo da izvršimo nakon što korisnik klikne na dugme. Prosleđujemo samo ime funkcije, ne kao poziv i bez argumenata.

Polje za izbor (checkbox)

Ova kontrola se koristi kada želimo da korisniku ponudimo da/ne izbor. Argumenti su:

- text – tekst opcije koju želimo da čekiramo;
- variable – argument koji prima tip tkinter promenljive BooleanVar ili IntVar u koju će se smeštati vrednost polja za izbor;
- textvariable – argument koji prima tip tkinter promenljive StringVar u koju se može smestiti tekst opcije koju želimo da čekiramo.

Ova kontrola je slična HTML tagu input tipa checkbox i inicijalizuje se naredbom tk.Checkbutton().

Pomoćna korišćenja checkbox vidžeta

```
import tkinter as tk
root = tk.Tk()
var = tk.IntVar()
chk = tk.Checkbutton(root, text='Choice 1', variable=var)
chk.pack()
tk.mainloop()
```

Napravite checkbutton-e za odabir pola.

Listbox kontrola

Ova kontrola je jedna od dve kontrole u Tkinteru koje se koriste za prikaz podataka u redovima. Druga kontrola je naprednija, ali i kompleksnija i zove se Treeview. Jedna od većih razlika između ove dve kontrole je u tome što ListBox ne podržava rad sa kolonama, već je moguće koristiti samo jednu kolonu.

Listbox kontrola se kreira pomoću tk.Listbox() klase. Iako podržava argumente koji su zajednički i većini drugih kontrola, kao što su boja i dimenzije, podržava i argument selectmode, čije vrednosti mogu biti tkinter.BROWSE, tkinter.SINGLE, tkinter.MULTIPLE i tkinter.EXTENDED, što predstavlja šta i koliko korisnik može obeležiti u kontroli.

Za razliku od nekih prethodnih kontrola, ova podržava veći set metoda koje se mogu pozvati nad Listbox objektom. Neke od bitnijih metoda su:

- curselection() – u program vraća trenutno obeležen red;
- activate(index) – obeležava (selektuje) red zadat indeksom;
- get(first, last = None) – dobavlja elemente iz liste u zadatom opsegu između first i last reda; argument last nije obavezan; ako se izostavi, ima podrazumevanu vrednost None; indeksi počinju od nule; ako prosledimo samo first argument, dobijamo jedan element;
- insert(index, *elements) – ubacuje element (ako ubacujemo više elemenata, onda taj parametar prosleđujemo kao n-torku); takođe, pored argumenta index koji prosleđujemo, možemo proslediti i konstante: tk.END – ubacujemo element na kraj liste i tk.ACTIVE – ubacujemo element umesto obeleženog elementa u listi.
- size() – predstavlja ukupan broj redova u listi.

Ostatak metoda se može pronaći u zvaničnoj [dokumentaciji](#).

Da bismo pokazali kako ova kontrola funkcioniše, možemo izmeniti prethodni primer i dodati funkcionalnost tako da nakon klika na dugme ispišemo tekst u Text kontroli, te isti taj tekst dodamo kao novi red u listi. Taj kod može izgledati ovako:

Primer upotrebe ListBox kontrole

```
import tkinter as tk
counter = 0
def button_func(text_widget, text_lines):
    global counter
    counter += 1
    text_widget.delete("1.0", tk.END)
    text_widget.insert("1.0", "Button clicked {}
times\n".format(str(counter)))
    text_lines.insert(tk.END, "Button clicked {}
times\n".format(str(counter)))
win = tk.Tk()
win.minsize(300, 350)

tk.Label(text = 'Text box:')
text_widget = tk.Text(master = win, height = 5, width = 25)

text_widget.pack()
tk.Label(text = 'List items:').pack()
text_lines = tk.Listbox(master = win, width = 30)

text_lines.pack()
tk.Button(master = win, text = 'ClickMe!', command = lambda:
button_func(text_widget, text_lines)).pack()

win.mainloop()
```

Labele su dodate radi lakšeg raspoznavanja kontrola. Kontrole Text i ListBox smo inicijalizovali sa unapred zadatim dimenzijama height/width. Vrednosti ovih argumenata se ne odnose na piksele, već na broj karaktera koji mogu stati po dužini i širini. Funkciju button_func() smo izmenili i dodali joj još jedan argument: kontrolu text_lines tipa tk.Listbox. Dodavanje u listu smo izvršili linijom text_lines.insert(tk.END, "Button clicked {} times\n".format(str(counter))), gde pomoću insert() metode na kraj liste (konstanta tk.END) dodajemo željeni string.

Izmenite kod tako da se u pomoću input funkcije unosi jedan ceo broj. Izmeniti funkciju button_func tako da se u listboxu ispisuje u svakom novom redu vrednost tog brojača povećana svaki put za vrednost istog. Npr. Ako se unese broj 3, prvim klikom bi trebalo da se ispiše u listboxu vrednost 3, u sledećem 6, pa 9...

Rezime

- Neki od najčešćih argumenata koji su dostupni skoro svim kontrolama su: background, foreground, font, cursor, padx, pady, width, height, borderwidth, text i font.
- Button kontrolu koristimo kada želimo da omogućimo direktnu interakciju korisnika sa programom. Samo dugme može sadržati, tj. prikazati tekst, ali i slike (ikonice).
- Text vidžet je sličan Entry kontroli, sa bitnom razlikom da je u Text vidžetu moguće uneti tekst sa više linija. Bitne metode su get(), insert() i delete().

- Polje za izbor (checkbox) koristi se kada korisniku želimo da ponudimo da/ne izbor. Argumenti prilikom inicijalizacije su text, variable i textvariable. Najsličnija je HTML tagu input tipa checkbox.
- ListBox kontrola je jedna od dve kontrole u Tkinteru koje se koriste za prikaz podataka u redovima. Druga kontrola je naprednija, ali i kompleksnija i zove se Treeview. Neke od bitnijih metoda za rad sa Listbox objektom su: curselection(), activate(), get(), insert() i size().



linkgroup